REMARKS

Claims 1-24 are now pending in the application. In an office action dated July 8, 2002 ("Office Action"), the Examiner noted a numerical labeling error in the specification, rejected claims 8 and 24 under 35 U.S.C. § 112, second paragraph, rejected claims 1-7 and 11-23 under 35 U.S.C. § 103(a) as being unpatentable over Wies et al., U.S. Patent No. 6,161,126 ("Weis"), rejected claims 8 and 24 under 35 U.S.C. § 103(a) as being unpatentable over Wies in further view of White et al., U.S. Patent No, 6,034,689 ("White"), and rejected claims 9-10 under 35 U.S.C. § 103(a) as being unpatentable over Wies, in further view of Kernz, U.S.Patent No. 6,366,899 B1 ("Kernz"). Applicants' representative has corrected the numerical mislabeling pointed out by the Examiner, and has amended claims 8 and 24 to overcome the Examiner's 35 U.S.C. § 112 rejection. Applicants' representative compliments the Examiner on a review of the application sufficiently careful and conscientious to notice these problems. Applicants' represent respectfully traverses the 35 U.S.C. § 103(a) rejections of claims 1-24.

Current Application

It is a rather tedious task to respond to the Examiner's 35 U.S.C. § 103(a) rejections, because they contain a great number of references to portions of Wies, White, and Kernz that disclose and suggest that which Applicants have acknowledged, in the current application, to be prior art. For example, in Figure 1 of the current application, Applicants show a web page with active regions (106 and 108). Applicants do not state or suggest that web pages with active regions are novel, nor do Applicants attempt to claim web-page-related technologies that existed at the time that they filed the current application. In fact, Applicants discuss, in detail, existing technologies for serving the exemplary web page shown in Figure 1 by a server to a client computer and displaying the exemplary web page to a human user on a client computer.

As discussed in the current application beginning on line 4 of page 6, at the time the current application was filed, the normal approach to displaying a web page, such as the exemplary web page shown in Figure 1, was to employ an HTML representation of the web page. An exemplary HTML encoding is shown starting at line 14 of page 7. The active regions are designated as such in AREA tag sections within a MAP definition, in terms of

device coordinates. When processed on a client computer, the HTML encoding results in an image, as defined in an image source file that is specified in an "img" tag on line 1 of the exemplary HTML encoding, over which a client-side image map is abstractly superimposed, as shown in Figures 3 and 4. As discussed starting on line 21 of page 8, certain enhanced web-page browsers support dynamic images, flagged as dynamic, in the case of OpenPix image, by an OPXVtype attribute as shown in exemplary HTML code beginning on line 6 or page 9. This allows the enhanced browser to locally provide various types of operations, including zoom operations, directed to a dynamic image, with out needing to exchange data with the server. Thus, as shown in Figure 5, an enhanced browser can zoom the image to an increased size. However, the client-side image map that maps active regions is not automatically expanded to track the change in image size. As a result, as shown in Figure 5, the client-side image map (504 in Figure 5) no longer properly superimposes onto, or corresponds to, the expanded image (502 in Figure 5). Figure 4 may be contrasted to Figure 5 to understand the problem. Again, all of this was known at the time that the current application was filed.

In one embodiment, Applicants' claimed invention addresses the type of problems illustrated in Figure 4 and 5 by providing for specification, in an HTML encoding of a web page, of an enhanced viewer for displaying a dynamic image, and specification of dynamic-adaptive client-side image maps ("DACSIMS") associated with the dynamic image. The client-side browser detects the invocation in the HTML code of an enhanced viewer, and instantiates the enhanced viewer to display the dynamic image, passing to the enhanced viewer parameters passed to the browser in the HTML encoding of the web page. It should be noted that the enhanced viewer is generally a separate thread or process that run concurrently with the browser on the client computer. The word "instantiate" in computer science implies creation of a process, thread, or object with associated, persistent state. As discussed beginning with the final paragraph on page 7 of the current application, the browser receives various events, including user inputs through keyboard and mouse devices, while the image is displayed, and passes the events to the enhanced viewer. The enhanced viewer maintains the data structures shown in Figure 8 to represent the image, and active regions within the image, and applies events to the image using information contained in the data structure.

An important feature of the system is that active-region coordinates are stored as relative coordinates, or, in other words, as fractional indices related to the image, rather than to the display device. These factional coordinates are described on page 17 with reference to the C++-like class "Point." By storing fractional coordinates to represent the active regions, the enhanced viewer does not need to constantly recalculate the positions of active regions based on device coordinates as the image is transformed by zoom, pan, and other operations, nor do long chains of object-relative coordinates need to be calculated and maintained. Instead, operations that change the size, shape, and position of the displayed image result in changing only the device coordinates of the image, rather than the changing both the device coordinates of the image and the device coordinates of all active regions. In essence, by using image-relative coordinates for active areas, the enhanced viewer efficiently stores the correspondence between the displayed image and all active regions within the displayed image in an image-position-and-image-size independent fashion, and need not constantly carry out complex recalculations of device-based active-region coordinates. Thus, as shown in the implementation of the Image function member "zoom," beginning at the bottom of page 23, the enhanced viewer only needs to change the device-based coordinates "leftX," "rightX," "lefty," and "rightY" for the image. Because all active regions are stored in image-relative coordinates, no update of active-region coordinates is required. When the enhanced viewer needs to determine whether or not a mouse input operation was input to an active region, the enhanced viewer simply transforms the device-coordinates of the mouse click point into image-relative coordinates and then iterates (as shown in the *while*-loop of lines 14-23 in the Image function member "mouseMove" on pages 25 and 26) through active regions to determine whether the image-relative coordinates of the mouse click fall within one or more active regions.

Consider claim 1, reproduce below:

1. A method for associating an active region with a corresponding position within an image included in a page displayed by a browser running on a client computer, the method comprising:

sending a request by the browser to a server for a description of a page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and specifying actions to be performed in response to input events directed to the active region;

receiving from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, the invocation including parameters that describe the image and the client-side image map;

instantiating the viewer and passing to the viewer the parameters included in the invocation;

storing by the viewer representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region; and

when an input event is detected by the browser during display of the page, passing the input event by the browser to the viewer, and

when the viewer determines that the input event was input to a position within the image corresponding to the active region, determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action. (emphasis added)

Thus, claim 1 clearly claims an enhanced viewer that runs along with a browser on the client side that is invoked by the browser with parameters received by the browser from the server. Claim 1 clearly claims storage of active-region representations in image-relative coordinates, as well as indications of actions to be performed in response to input events directed to the active regions. The browser fields input events, and passes them to the enhanced viewer, and the enhanced viewer then applies the events to any active region to which the event was directed.

Similarly, consider claim 11:

11. A method for serving a description of a page from a server to a browser running on a client computer that requests the page, the description of the page provided to the browser by the server containing an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image, the method comprising:

receiving a request from the browser by the server for a description of the page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and that specifies actions to be performed in response to input events directed to the active region;

retrieving a description of the page;

determining the capabilities for viewing pages provided by the browser running on the client computer, and

when the browser running on the client computer is capable of accepting display altering commands from a user while displaying a page,

parsing the description of the page to find the specification of the image and the client-side image map included in the page,

substituting, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page, including in the invocation parameters that specify the image and the client-side image map, to create a transformed page description, and

sending the transformed page description to the browser. (emphasis added)

As clearly claims in claim 11, the server receives a request for a page from a client and determines the capabilities for viewing pages provided by the browser from the request, and, when the client is capable of invoking an enhanced viewer to display dynamic images, encodes substitutes an invocation of the enhanced viewer into an encoding of the web page in place of an image specification, along with parameters that specify to the enhanced viewer the image and a client-side image map. The server then sends the transformed encoding of the web page to the client.

<u>35 U.S.C. § 103(a) Rejections based on Wies</u>

Wies is not directly related to Applicants' claim invention. Wies concerns providing a force-feedback mechanism during display of web pages on a client computer. In essence, this means that, when the client manipulates a mouse or other device during the display of the web page, Wies' system transfers force-feedback information to the client computer for input to the mouse or other device.

Applicants' representative cannot understand the Examiner's arguments with respect to claim 1. The Examiner states that Wies discloses "sending a request by the browser to a server for a description of a page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region" in column 6, lines 27-46, column 4, lines 23-59, column 12, lines 55-67, figure 10, and figures 13a-b, 14, 17a-b, and column 26, lines 4-39. In column, lines 27-46, Wies generically describes sending an HTML encoding of a web page from a server to a client, and providing force-feedback to a client during display of the web page on the client. In column 4, lines 23-59, Wies generically suggests associating forces, <u>by a human user using a web-page authoring tool</u>, with specific parts of a web page, and encoding the force information in the web page, along with sounds. In column 12, lines 55-67, Wies generically describes associating force effects with objects displayed on a web page. In Figure 10, Wies *discloses operation of a browser* – not an enhanced viewer running concurrently with a browser, as claimed in claim 1 - *invoked by a human user* on a client computer – not by a browser receiving an encoded invocation in an HTML encoding, as claimed in claim 1 – which directly determines, by *calculating screen coordinates*, when a mouse pointer is positioned in relation to a relevant web object – not by using image-relative

coordinates, as claimed in claim 1. Wies clearly annotates Figure 10 on lines 12-33 of column 22:

> As shown in a method 320 of FIG. 10, the user again starts an application program on the client machine in step 322. In step 324, this application program opens a browser window ... the application program then applies then applies the methodology described above in FIG. 7 ... to directly determine (by calculating screen coordinates) when the mouse pointer is positioned in relation to a relevant web object ... such that screen coordinates are calculated ... and forces are to be output to cause the force feedback interface device to output the appropriate forces ... (emphasis added)

Figures 13a-b, 14, and 17a-b illustrate a simple force-feedback application, displayed by a browser invoked by a human user, and, in column 26, lines 4-39, Wies discusses this simple force-feedback application. These cited sections disclose that a human user may author a web page that includes a specification of forces to apply to a user device during viewing of the web page, but they nowhere indicate how this is accomplished. There is no indication of a client-side image map, as described in detail in the current application, and clearly claimed in claim 1. Apparently, the Examiner assumes that a client-side image map is included in the authored web page, but there is no indication in Wies that a client-side image map is used.

Next, the Examiner states that Wies discloses "receiving from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, the invocation including parameters that describe the image and the client-side image map" in lines 4-56 of column 26 and column 28, line 20 to line 21 of column 29. In lines 4-56 of column 26, Wies describes a force feedback application program. In the section starting with line 20 of column 28 and extending to line 21 of column 29, Wies describes the "<EMBED > tag that defines a force button object. As stated by Wies in this section, "The <EMBED ...> command is an existing functionality of HTML. It essentially embeds function calls which are handled by the web browser." Wies further points out that the web browser may call a DLL or other application program to carry out the function if it is not capable of doing so itself. But, it should be noted that calling functions implemented in an application program or DLL is not the same thing as instantiating a separately executing viewer which stores representations of active regions and to which the browser passes events that occur over a period of time, for handling by the viewer. Also note

that, unlike Applicants' claimed invention, the browser in Wies determines which function in which DLL or application to call, when it cannot itself execute the function. A function call executes to completion in the address space of the calling program, while execution of the calling program is suspended. The viewer instantiated in Applicants' claimed system runs concurrently with a browser, and is passed events by the browser on an on-going basis.

Next, the Examiner states that Wies discloses "storing by the viewer representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region" in the section starting at line 65 of column 31 to line 23 of column 32. Perhaps the Examiner has not understood the term "viewer" in Applicants' claim language. The cited portion of Wies refers to inputs *that a human user may make to an interactive authoring tool.* The viewer in Applicants' claims refers to a process or thread running on the client computer that executes concurrently with a browser. Note also that the viewer in Applicants' system stores representations of active regions in image-relative coordinates. This is described, above, with reference to relevant sections of the current application. The cited section of Wies has nothing to do with Applicants' claimed viewer. Image-relative coordinates are an important, and explicitly claimed feature of Applicants' system – Wies does not use, mention, or suggest image-relative coordinates, but instead, as discussed above, specifically discloses using device or screen coordinates. Note that Wies specifically mentions one problem addressed by Applicants in the cited portion beginning on line 54 of column 20 and extending to line 22 of column 21 – namely the "difficulty of this approach lies in determination of screen coordinates of the relevant object." However, as Wies clearly describes in the cited portion, and clearly shows in Figures 8 and 9, Wies constantly calculates screen coordinates based on a complex chain of nested offsets describing the relative positions of objects with respect to enclosing objects. Applicants' clearly described and clearly claimed image-relative coordinates are quite different – they are fractional coordinates based solely on the outermost image. There are no chains of offsets calculated by Applicants' viewer.

Next, the Examiner states that Wies discloses "passing the input event by the browser to the viewer when the input event is detected by the browser during display of the page" in the section starting at line 65 of column 31 to line 23 of column 32. Again, the Examiner apparently has mistaken Applicants' claimed viewer, a separate process or thread

invoked by a browser to run concurrently with the browser, with a human user. Nowhere does Wies suggest or mention invoking a viewer by a browser and continuously passing input events detected by the browser to the viewer during display of a web page.

Finally, the Examiner states that Wies discloses "determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action when the viewer determines that the input event was input to a position within the image corresponding to the active region" in figures 13a-b, 14, 17a-b, and column 26, lines 4-39. As discussed above, Wies does not mention or suggest a viewer thread or process running concurrently with a web browser that detects when input events are directed to active regions and applies specified actions. The cited portions of Wies merely describe operation of a force feedback application.

The Examiner then states that "Wies does not disclose explicitly a client-side image map including an active region and associated with the image of the requested page." The Examiner then concludes that it would have been obvious to transform Wies' force feedback application into Applicants' completely dissimilar, claimed dynamic web page displaying system.

Applicants' clearly claim an enhanced viewer that runs concurrently with a browser on a client computer to monitor input events and carry out any events directed to active regions. Applicants' enhanced viewer stores a representation of active regions, in image-relative coordinates, in a client-side image map. Wies' force feedback application does not invoke a viewer process or thread. Wies does not suggest or mention a client-side image map with active regions represented in image-relative coordinates. Applicants' clearly claimed method bears virtually no similarity to the rather traditional, web-page display method employed by Wies. The Examiner has failed to find any teaching, mention, or suggestion in Wies for these clearly claimed elements, and, in order to make a 35 U.S.C. § 103(a) rejection, the Examiner must identify a teaching, mention, or suggestion of each claimed element. Therefore, Applicants' representative believes that claim 1 cannot possibly be obvious in view of Wies.

With regard to claim 11, the Examiner lists 5 separate elements as not having been disclosed by Wies, but states that claim 11 would be obvious in view of Wies since Wies discloses "sending an appropriate web page to the requesting client." Applicants'

representative reads this statement with incredulity. Note that claim 11 specifically claims "substituting, in the description of the page, an invocation of a viewer for the specification of the image" and "including in the invocation parameters that specify the image and the client-side image map." As established above, Wies neither mentions nor suggests a viewer, does not mention of suggest invocation of a viewer, and does not mention or suggest a client-side image map. Wies' force feedback system bears no similarity to Applicants' claimed invention. As with claim 1, claim 11 cannot possibly be obvious in view of Wies, since Wies fails to mention or suggest almost every claimed element. A similar, tedious analysis may be offered for independent claim 18 and all dependent claims. In view of the rather enormous disparity between Wies' disclosed force feedback application and Applicants' clearly claimed method and system, Applicants' representative, in the interest of brevity, omits the largely redundant arguments to support traversal of all 35 U.S.C. § 103(a) rejections, relying on the arguments made with respect to claim 1, above. Applicants' representative notes, however, that Kernz and the cited passages and figures of White appear to be as equally irrelevant as Wies with regard to Applicants' claimed invention.

Applicants' representative hopes that the Examiner will re-read the current application, in light of the concise summary enclosed above, to re-evaluate the Examiner's 35 U.S.C. § 103(a) rejections. Applicants' representative believes that all of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,
Lawrence Bain et al.
Olympic Patent Works, PLLC

Robert W. Bergstrom
Robert W. Bergstrom
Registration No. 39,906

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98104
206.621.1933 telephone
206.621.5302 fax

Version With Markings to Show Changes Made

In the Specification:

The paragraph beginning on Page 7, line 6, is amended as follows:

Figure 2 is an expanded view of the smaller complex image component of the web page displayed in Figure 1. The image can be described in terms of device coordinates correlated to the pixel density of the client computer display monitor on which the web page shown in Figure 1 is displayed. The origin of the device coordinate system, with coordinates (0,0), occurs at the upper left hand corner of the displayed web page 202. An *x* axis 204 emanates horizontally from the origin, and a *y* axis 206 emanates downward from the origin. The width of the image [208] 204 shown in Figure 2 is 116 unites, where units roughly correspond to pixels, and the height of the image is 450 units. Thus, in terms of the device coordinate system, the four corners 202, 210, 212, and 214 of the image shown in Figure 2 have coordinates (0,0), (116,0), (116,450), and (0,450), respectively.

In the Claims:

Claims 8 and 24 are amended as follows:

8. (Amended) The method of claim 2 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).

24. (Amended) The method of claim 19 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of

coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).